Hands-on Workshop: Introduction To MPI For Finite Difference Models

Luke Lonergan
High Performance Technologies, Inc.

Course Strategy

In this course we will cover the following topics:

- MPI Basic Functions and Their Use
- A Prototype Finite Difference Model
- Serial Code Version of the Model
- A 1-D MPI Decomposition Version of the Model
- Basic MPI Profiling
- Input/Output for Distributed Programs
- A 2-D MPI Decomposition Version of the Model

MPI Basic Functions and Their Use

- We will use the six major functions of MPI to develop some simple programs, giving us a feel for the approach to writing MPI codes.
- We will be using the computers in the lab as a testbed for the MPI programs we write. These programs can be executed on any machine with MPI support.
  - The lab includes 19 computers linked by a 5 MB/s Ethernet network
  - 8 (7 up) Dual Pentium Pro 200s - Linux
  - 11 (9 up) SGI O2 R10000s - IRIX 6.3

A Prototype Finite Difference Model

We will introduce a finite difference model that will serve to demonstrate what a computational scientist needs to do to take advantage of Distributed Memory computers using MPI

The model we are using is a two dimensional solution to a model problem for Oceanography

Serial Code Version of the Model

We will develop the basic serial (not parallel) version of the finite difference model

The emphasis will be on covering the basic features of the model and code in enough detail so that the transformations used in the MPI implementation are clearly understood

A 1-D MPI Decomposition

We will choose one of the two dimensions and subdivide the domain to allow the distribution of the work across a group of distributed memory processors

We will focus on the principles and techniques used to do the MPI work in the model and characterization of the performance of the model

Basic MPI Profiling

We will cover the process of profiling the performance of an MPI code using simple profiling tools for message passing

This exercise will cover the characterization of the overheads associated with communication among processors

A 2-D MPI Decomposition

Here we will develop the 2-D decomposition of the model, using the same concepts developed in doing the 1-D version

The goal of the exercise is to show how the effort of doing the multidimensional message passing work can be simplified using the topological functions in MPI

Distributed Input/Output

We will develop a scheme for distributing the input and output of files for our prototype model

The emphasis here will be on getting some practical experience with one of the more difficult topics in parallel computing: Parallel I/O

Administrivia - Logging on, etc.

- Set up the remote X display
  - Find local IP address of your screen (local_ip)
  - On local window, type: “xauth list”
    - Grab everything with the left mouse _after_ the pet0x...
  - On remote window type: “setenv DISPLAY local_ip:0”
  - On local window type: “xauth add $DISPLAY {mouse paste}”